AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application:

1. (currently amended) A method for hashing, comprising:

storing a plurality of partial keys corresponding to a plurality of original keys in memory, wherein storage of said plurality of partial keys requires less memory than storage of said plurality of original keys, and wherein said plurality of partial keys are used to determine hashing conflicts;

applying a hash function to an original key of said plurality of original keys to generate a hash value, wherein said hash function comprises any polynomial;

accessing the memory according to the hash value;

reading a partial key of said plurality of partial keys from the memory that corresponds to the said hash value, wherein said hash value is based on said original key;

executing a conflict check by comparing a partial key derived from an incoming full key with the partial key stored in the memory.

2. (previously presented) The method of Claim 1, wherein the partial key from the memory corresponding to the hash value includes saved bits comprising a consecutive, sequential string of bits, less than or equal to N, which is part of the original key.

3. (previously presented) The method of Claim 2, wherein the partial key from the memory corresponding to the hash value comprises a number of bits equal to or more than the number of bits of the original key minus the number of bits of the hash value.

4. (original) The method of Claim 1, wherein the hash function is implemented by a linear feedback shift register.

5. (previously presented) The method of Claim 1 further comprising applying a reverse function on the partial key from the memory corresponding to the hash value and hash value to generate the original key.

6. (original) The method of Claim I further comprising the steps of:

reading a result from the memory corresponding to the hash value;

forwarding a packet of data according to the result read from the memory.

7. (currently amended) An apparatus A hashing apparatus, comprising:

a memory which stores a plurality of partial keys used to determine hashing conflicts, wherein said plurality of partial keys correspond to a plurality of original keys, and wherein storage of said plurality of partial keys requires less memory than storage of said plurality of original keys;

a hash function block coupled to a memory that applies any polynomial to a full key and generates a hash value which is used to point to one of the plurality of partial keys stored in the memory, wherein the partial keys include saved bits comprising a consecutive, sequential string of bits derived from the original key.

8. (currently amended) The hashing apparatus of Claim 7, wherein the memory comprises a hash table size.

9. (currently amended) The hashing apparatus of Claim 7, wherein the one of the plurality of partial keys stored in the memory comprises a number of bits equal to or more than the number of bits of the original key minus the number of bits of the hash value.

10. (currently amended) The hashing apparatus of Claim 7, wherein the hash function block comprises a linear feedback shift register.

11. (currently amended) The hashing apparatus of Claim 9, wherein the

linear feedback shift register corresponds to a Galois version.

12. (currently amended) The ~~hashing~~ apparatus of Claim 9, wherein the linear feedback shift register corresponds to a Fibonacci version.

13. (currently amended) The ~~hashing~~ apparatus of Claim 7 further including a reverse function generator coupled to the memory, wherein the reverse function generator generates the original key based on the one of the plurality of partial keys stored in the memory and hash value.

14. (currently amended) The ~~hashing~~ apparatus of Claim 7 further comprising a forwarding engine coupled to the memory, wherein the forwarding engine forwards a data packet according to information read from the memory at an address corresponding to the one of the plurality of partial keys stored in the memory.

15. (currently amended) An apparatus ~~A hashing apparatus~~ comprising:
means for storing a plurality of partial keys corresponding to a plurality of original keys in memory, wherein storage of said plurality of partial keys requires less memory than storage of said plurality of original keys, and wherein said plurality of partial keys are used to determine hashing conflicts;
means for applying a hash function to an original key of said plurality of original keys to generate a hash value, the hash function comprising any N bit polynomial;
means for accessing the memory according to the hash value, wherein a position to save comprises any N consecutive bits;
means for reading a partial key of said plurality of partial keys from the memory corresponding to the hash value and based on the original key, wherein a size to save comprises ≤ N bits;
means for executing a conflict check by comparing a partial key derived from an incoming full key with the partial key stored in the memory, wherein the

memory comprises a 2^N hash table size.

16. (currently amended) The ~~hashing~~ apparatus of Claim 15, wherein saved bits comprise a consecutive, sequential string of bits which is part of the original key.

17. (currently amended) The ~~hashing~~ apparatus of Claim 16, wherein the partial key from the memory corresponding to the hash value comprises a number of bits equal to or more than the number of bits of the original key minus the number of bits of the hash value.

18. (currently amended) The ~~hashing~~ apparatus of Claim 15, wherein the hash function is implemented by a linear feedback shift register means.

19. (currently amended) The ~~hashing~~ apparatus of Claim 15 further comprising means for applying a reverse function on the partial key from the memory corresponding to the hash value and hash value to generate the original key.

20. (currently amended) The ~~hashing~~ apparatus of Claim 15 further comprising:
    means for reading a result from the memory corresponding to the hash value; means for forwarding a packet of data according to the result read from the memory.

21. (canceled)

22. (canceled)

23. (canceled)

24. (canceled)

25. (canceled)

26. (currently amended) A method ~~for accessing data,~~ comprising:

storing a plurality of partial keys <u>corresponding to a plurality of original</u> <u>keys</u> in memory, <u>wherein storage of said plurality of partial keys requires less</u> <u>memory than storage of said plurality of original keys, and wherein said plurality</u> <u>of partial keys are used to determine hashing conflicts</u>;

applying a function to an original key <u>of said plurality of original keys</u> to generate a value;

accessing the memory according to the value:

reading a partial key <u>of said plurality of partial keys</u> from the memory corresponding to the value and based on the original key;

comparing the partial key to the value in determining which data is accessed;<u> and</u>

applying a reverse function on the partial key and value to generate the original key.